

1. Ensimmäiset mikroprosessorit olivat ns. akkukoneita. Niitä kutsutaan myös yksioperandikoneiksi. Piirrä kaaviokuva tällaisesta prosessorista ja selitä sen toimintaperiaate. Kerro, kuinka rakennetta voidaan parantaa (esim. Intelin Pentium ei tunnetusti ole akkukone).

2. Yksinkertainen 8-bittinen akkuprosessori suorittaa peräkkäin käskyt LDI 42h ja STA 8012h. Ensimmäinen käsky (LDI 42h) lataa akkuun välittömän vakion 42h. Toinen käsky tallettaa akun arvon muistiosoitteeseen 8012h. Piirrä väylän ajoituskaavio käskyjen suorituksen ajalta. Kuinka suuri saa enintään olla muistin haku-aika jotta mainitun käskysarjan suoritus kestäisi yhden mikrosekunnin jos oletetaan, että ohjelma- ja datamuistit ovat yhtä nopeita eikä muita suorituksen nopeutta rajoittavia tekijöitä ole?

3. Mihin tietokonejärjestelmä tarvitsee muistia? Kerro muistityypeistä, joita järjestelmäsuunnittelijalla on käytettävissä ja mihin tarkoitukseen ja minkälaisessa järjestelmässä kutakin käyttäisit.

4. Suunnittele mielestäsi optimaalinen muistikartta tietokoneelle, joka voi toteuttaa tehtävän 2 toiminnan. Jätä tilaa I/O-piireille. Valitse sopivat muistipiirit ja tee muistikarttaasi vastaava osoitteenkoodaus.

Vastaa vain jompaankumpaan tehtävistä 5 tai 6.

5. Jatka siitä, mihin jäit tehtävässä 4 ja piirrä koko kone. Koneessa täytyy olla sisään- ja ulostuloja.

6. Jos ajatellaan, että tietokonetekniikan historian lehtien rapinaa rapisee yksioperandi- eli akkukone, jossa aritmeettisen käskyn toinen operandi on aina akku ja toinen on välitön vakio tai muistiosoitteeseen niin sitten huippunykyaikaista tekniikkaa olisi ns. load-store -arkkitehtuuri, jossa aritmetiikka tapahtuu rekistereissä, joita on monta, ja muistiin pääsee ainoastaan load ja store -käskyillä, jotka hakevat ja tallettavat rekisterin arvon muistiosoitteeseen (osoite sijaitsee myös rekisterissä). Parhaat load-store -koneet ovat kolmioperandisia eli esim. käsky `add r1, r2, r3` tallettaisi rekisterien `r2` ja `r3` summan `r1`:een.

Muistiosoitteissa `4000h-401fh` sijaitsee taulukko, jonka alkiot ovat sanan levyisiä (eli ensimmäinen alkiio sijaitsee osoitteessa `4000h`, seuraava osoitteessa `4001h` jne.) Suunnittele molemmille arkkitehtuureille rutiini, joka laskee taulukoiden arvojen summan. (sananleveydellä ei ole tehtävän kannalta merkitystä). Pyri mahdollisimman pieneen rutiiniin.

Akkuprosessorissa on aritmeettinen rekisteri `a` (akku) ja osoiterekisteri `x` eikä muita käytettävissä olevia rekistereitä. Osoiterekisteriin voi ladata vakion ja sen arvoja voi lisätä ja vähentää `inc` ja `dec` -käskyillä, mutta `x`-rekisterin arvoa ei voi lukea joten et voi `x`-rekisterin arvoista tutkia oletko päässyt alueen loppuun. Tarvitset tilapäismuuttujia muistissa. Aritmetiikassa esim. käsky `add x` lisää akkuun muistissa osoiterekisterin `x` osoittamassa osoitteessa olevan arvon.

Load-store -prosessorissa on 32 yleisrekisteriä. Mitä tahansa rekisteriä voi käyttää data- tai osoiterekisterinä. Esim. `load r1, r2` hakee muistista rekisterin `r2` osoittamasta osoitteesta arvon rekisteriin `r1`.

Vertaile lopuksi sanallisesti tuloksiasi ja kerro, minkälainen mielikuva Sinulle jäi kummastakin ohjelmointitavasta.

Lämmintä kesää kaikille!  
terv. Panze.